```
LLL                  IIIIIIIIII   BBBBBBBBBBBB   RRRRRRRRRRR    TTTTTTTTTTTTTTT  LLL
LLL                  IIIIIIIIII   BBBBBBBBBBBB   RRRRRRRRRRR    TTTTTTTTTTTTTTT  LLL
LLL                  IIIIIIIII    BBBBBBBBBBBB   RRRRRRRRRRR    TTTTTTTTTTTTTTT  LLL
LLL                     III       BBB      BBB   RRR      RRR        TTT         LLL
LLL                     III       BBB      BBB   RRR      RRR        TTT         LLL
LLL                     III       BBB      BBB   RRR      RRR        TTT         LLL
LLL                     III       BBB      BBB   RRR      RRR        TTT         LLL
LLL                     III       BBB      BBB   RRR      RRR        TTT         LLL
LLL                     III       BBBBBBBBBBBB   RRRRRRRRRRR         TTT         LLL
LLL                     III       BBBBBBBBBBBB   RRRRRRRRRRR         TTT         LLL
LLL                     III       BBB      BBB   RRR   RRR           TTT         LLL
LLL                     III       BBB      BBB   RRR   RRR           TTT         LLL
LLL                     III       BBB      BBB   RRR    RRR          TTT         LLL
LLL                     III       BBB      BBB   RRR     RRR         TTT         LLL
LLL                     III       BBB      BBB   RRR     RRR         TTT         LLL
LLLLLLLLLLLLLLLL        III       BBBBBBBBBBBB   RRR      RRR        TTT         LLLLLLLLLLLLLLLL
LLLLLLLLLLLLLLLL     IIIIIIIII    BBBBBBBBBBBB   RRR      RRR        TTT         LLLLLLLLLLLLLLLL
LLLLLLLLLLLLLLLL     IIIIIIIII    BBBBBBBBBBBB   RRR      RRR        TTT         LLLLLLLLLLLLLLLL
```

```
LL        IIIIII   BBBBBBBB    AAAAAA    SSSSSSSS  TTTTTTTTTT  IIIIII   NN      NN  PPPPPPPP
LL        IIIIII   BBBBBBBB    AAAAAA    SSSSSSSS  TTTTTTTTTT  IIIIII   NN      NN  PPPPPPPP
LL          II     BB    BB   AA    AA   SS            TT        II     NN      NN  PP    PP
LL          II     BB    BB   AA    AA   SS            TT        II     NN      NN  PP    PP
LL          II     BB    BB   AA    AA   SS            TT        II     NNNN    NN  PP    PP
LL          II     BB    BB   AA    AA   SS            TT        II     NN NN   NN  PP    PP
LL          II     BBBBBBBB   AA    AA   SSSSSS        TT        II     NN  NN  NN  PPPPPPPP
LL          II     BBBBBBBB   AA    AA   SSSSSS        TT        II     NN   NN NN  PPPPPPPP
LL          II     BB    BB  AAAAAAAAAA      SS        TT        II     NN    NNNN  PP
LL          II     BB    BB  AAAAAAAAAA      SS        TT        II     NN    NNNN  PP
LL          II     BB    BB   AA    AA       SS        TT        II     NN      NN  PP
LLLLLLLLLL  IIIIII BBBBBBBB   AA    AA   SSSSSSSS      TT      IIIIII   NN      NN  PP
LLLLLLLLLL  IIIIII BBBBBBBB   AA    AA   SSSSSSSS      TT      IIIIII   NN      NN  PP
```

```
LL        IIIIII   SSSSSSSS
LL        IIIIII   SSSSSSSS
LL          II     SS
LL          II     SS
LL          II     SS
LL          II     SS
LL          II     SSSSSS
LL          II     SSSSSS
LL          II         SS
LL          II         SS
LL          II         SS
LLLLLLLLLL  IIIIII SSSSSSSS
LLLLLLLLLL  IIIIII SSSSSSSS
```

```
0000    1            .TITLE  LIB$AST_IN_PROG - Inquire if AST in progress
0000    2            .IDENT  /1-004/                   ; File: LIBASTINP.MAR Edit DG1004
0000    3
0000    4    ;
0000    5    ;*******************************************************************************
0000    6    ;*                                                                             *
0000    7    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                    *
0000    8    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                     *
0000    9    ;*  ALL RIGHTS RESERVED.                                                       *
0000   10    ;*                                                                             *
0000   11    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED      *
0000   12    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE       *
0000   13    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER       *
0000   14    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY       *
0000   15    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY        *
0000   16    ;*  TRANSFERRED.                                                               *
0000   17    ;*                                                                             *
0000   18    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE       *
0000   19    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT       *
0000   20    ;*  CORPORATION.                                                               *
0000   21    ;*                                                                             *
0000   22    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS      *
0000   23    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                     *
0000   24    ;*                                                                             *
0000   25    ;*                                                                             *
0000   26    ;*******************************************************************************
0000   27    ;
0000   28
0000   29    ;++
0000   30    ; FACILITY: General Utility Library
0000   31    ;
0000   32    ; ABSTRACT:
0000   33    ;
0000   34    ;       Return SUCCESS (1) if an AST is active in the current mode.
0000   35    ;
0000   36    ; ENVIRONMENT: All Modes, AST Reentrant
0000   37    ;
0000   38    ;--
0000   39    ; AUTHOR: Jonathan M. Taylor, CREATION DATE: 23-Jan-78
0000   40    ;
0000   41    ; MODIFIED BY:
0000   42    ;
0000   43    ;       JMT, 23-Jan-78 : VERSION 0
0000   44    ; 01    - Original
0000   45    ; 1-001 - Update version number and copyright notice.  The edit number
0000   46    ;           on version 0 was 4.  JBS 16-NOV-78
0000   47    ; 1-002 - Add """" to the PSECT directive.  JBS 21-DEC-78
0000   48    ; 1-003 - Allocāte an event flag number for the $GETJPI using LIB$GET_EF.
0000   49    ;           SBL 24-Nov-1981
0000   50    ; 1-004 - Use $GETJPIW to ensure synchronous operation.  DG 26-Oct-1983
```

```
        0000    52              .SBTTL  DECLARATIONS
        0000    53
        0000    54  ;
        0000    55  ; EXTERNAL SYMBOLS:
        0000    56  ;
        0000    57              .DSABL  GBL
        0000    58              .EXTRN  LIB$STOP
        0000    59              .EXTRN  LIB$GET_EF
        0000    60              .EXTRN  LIB$FREE_EF
        0000    61              .EXTRN  LIB$_FATERRLIB
        0000    62
        0000    63  ;
        0000    64  ; INCLUDE FILES:
        0000    65  ;
        0000    66
        0000    67  ;
        0000    68  ; MACROS:
        0000    69  ;
        0000    70              $JPIDEF                              ; define system service constants
        0000    71
        0000    72  ;
        0000    73  ; EQUATED SYMBOLS:
        0000    74  ;    none
        0000    75
        0000    76  ;
        0000    77  ; OWN STORAGE:
        0000    78  ;    none
        0000    79
        0000    80  ;
        0000    81  ; PSECT DECLARATIONS:
        0000    82  ;
    00000000    83              .PSECT  _LIB$CODE PIC, SHR, LONG, EXE, NOWRT
```

F 11

```
                        0000    85              .SBTTL  LIBSAST_IN_PROG - is AST in progress?
                        0000    86
                        0000    87  ;++
                        0000    88  ; FUNCTIONAL DESCRIPTION:
                        0000    89  ;
                        0000    90  ;     Call system service GETJPI to get access modes with active
                        0000    91  ;     ASTs.  Pick off the bit corresponding to the current-mode
                        0000    92  ;     (taken from the PSL) and return it.
                        0000    93  ;
                        0000    94  ; CALLING SEQUENCE:
                        0000    95  ;
                        0000    96  ;     at_ast_level.wl.v = LIBSAST_IN_PROG ([efn.rl.r])
                        0000    97  ;
                        0000    98  ; INPUT PARAMETERS:
                        0000    99  ;
          00000004      0000   100  ;     efn = 4              ; Longword event flag number to use for $GETJPI,
                        0000   101  ;                          ; passed by reference. Optional. If omitted,
                        0000   102  ;                          ; LIB$GET_EF is called to allocate a number.
                        0000   103  ;
                        0000   104  ; IMPLICIT INPUTS:
                        0000   105  ;
                        0000   106  ;     NONE
                        0000   107  ;
                        0000   108  ; OUTPUT PARAMETERS:
                        0000   109  ;
                        0000   110  ;     NONE
                        0000   111  ;
                        0000   112  ; IMPLICIT OUTPUTS:
                        0000   113  ;
                        0000   114  ;     NONE
                        0000   115  ;
                        0000   116  ; FUNCTION VALUE:
                        0000   117  ;
                        0000   118  ;     Returns SUCCESS (1) if we're currently as AST level, otherwise 0.
                        0000   119  ;
                        0000   120  ;
                        0000   121  ; SIDE EFFECTS:
                        0000   122  ;
                        0000   123  ;     May signal LIB$_FATERRLIB
                        0000   124  ;
                        0000   125  ;--
                0000    0000   126          .ENTRY  LIBSAST_IN_PROG, ^M<>
          7E    D4      0002   127          CLRL    -(SP)                           ; set up a one-item item-list
          7E    7C      0004   128          CLRQ    -(SP)
       08 AE    DF      0006   129          PUSHAL  8(SP)                           ;ITMLST: _0300_|__4___|
 03000004 8F    DD      0009   130          PUSHL   #<JPI$_ASTACTa16>+4             ;                *
                        000F   131
                        000F   132                                                  ;                0
                        000F   133
       51 5E    D0      000F   134          MOVL    SP, R1                          ;                0
          6C    95      0012   135          TSTB    (AP)                            ; Zero arguments?
          1E    13      0014   136          BEQL    GET_EF                          ; Yes, call LIBSGET_EF
       04 AC    D5      0016   137          TSTL    efn(AP)                         ; Omitted by reference?
          19    13      0019   138          BEQL    GET_EF                          ; Yes, call LIBSGET_EF
                        001B   139          $GETJPIW_S      EFN=aefn(AP), ITMLST=(R1)
       4D 50    E9      002F   140          BLBC    RO, ERROR                       ; Signal if error
          35    11      0032   141          BRB     OK                              ; Skip to common processing
```

G 11

```
                              0034   142
                              0034   143 ;+
                              0034   144 ; Come here if we need to allocate an event flag number
                              0034   145 ;-
                              0034   146 GET_EF:
                 7E   D4      0034   147         CLRL    -(SP)                        ; Make a place for the event flag number
                 6E   9F      0036   148         PUSHAB  (SP)                         ; Address of efn
    00000000'GF  01   FB      0038   149         CALLS   #1, G^LIB$GET_EF             ; Get an efn
              3D 50   E9      003F   150         BLBC    R0, ERROR                    ; Exit if unsuccessful
                 50   5E  D0  0042   151         MOVL    SP, R0                       ; Move address of EFN
                              0045   152         $GETJPIW_S       EFN=(R0), ITMLST=(R1)
              24 50   E9      0058   153         BLBC    R0, ERROR                    ; branch if not successful
                 6E   9F      005B   154         PUSHAB  (SP)                         ; Address of efn
    00000000'GF  01   FB      005D   155         CALLS   #1, G^LIB$FREE_EF            ; Free the efn
              18 50   E9      0064   156         BLBC    R0, ERROR                    ; Signal if error
                 00   11      0067   157         BRB     OK                           ; Skip to finish up
                              0069   158
                              0069   159 ;+
                              0069   160 ; Come here for final processing
                              0069   161 OK:
           50    FC AD  D0    0069   162         MOVL    -4(FP), R0                   ; R0 = bitvector by mode
                 51   DC      006D   163         MOVPSL  R1
        51  51   08  9C       006F   164         ROTL    #8, R1, R1
           51    FC 8F  8A    0073   165         BICB    #-4, R1                      ; R1<0:7> = current mode
           51    51   8E      0077   166         MNEGB   R1, R1                       ; R1 = 0,-1,-2,-3
        50  50   51  9C       007A   167         ROTL    R1, R0, R0                   ; get proper flag into lsb
                 04           007E   168         RET
                              007F   169
                              007F   170 ERROR:
                 50   DD      007F   171         PUSHL   R0                           ; Push original error code
                 7E   D4      0081   172         CLRL    -(SP)                        ; Zero FAO parameters
    00000000'8F  DD           0083   173         PUSHL   #LIB$_FATERRLIB
    00000000'GF  03   FB      0089   174         CALLS   #3, G^LIB$STOP               ; Can't return
                 04           0090   175         RET                                 ; But have signalled PC in this module
                              0091   176
                              0091   177         .END
```

```
$$T1                       = 00000001
EFN                        = 00000004
ERROR                        0000007F R    02
GET_EF                       00000034 R    02
JPI$_ASTACT                = 00000300
LIB$AST_IN_PROG              00000000 RG   02
LIB$FREE_EF                  ******** X    00
LIB$GET_EF                   ******** X    00
LIB$STOP                     ******** X    00
LIB$_FATERRLIB               ******** X    00
OK                           00000069 R    02
SYS$GETJPIW                  ******** G    02
```

+-------------------+
! Psect synopsis !
+-------------------+

PSECT name	Allocation		PSECT No.	Attributes											
. ABS .	00000000 (	0.)	00 (0.)	NOPIC	USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE
ABS	00000000 (	0.)	01 (1.)	NOPIC	USR	CON	ABS	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE
_LIB$CODE	00000091 (	145.)	02 (2.)	PIC	USR	CON	REL	LCL	SHR	EXE	RD	NOWRT	NOVEC	LONG

+-----------------------------+
! Performance indicators !
+-----------------------------+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.05	00:00:00.81
Command processing	113	00:00:00.28	00:00:03.53
Pass 1	133	00:00:01.19	00:00:07.03
Symbol table sort	0	00:00:00.08	00:00:00.35
Pass 2	47	00:00:00.36	00:00:01.97
Symbol table output	3	00:00:00.01	00:00:00.01
Psect synopsis output	2	00:00:00.01	00:00:00.01
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	329	00:00:01.99	00:00:13.72

The working set limit was 900 pages.
8430 bytes (17 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 120 non-local and 0 local symbols.
177 source lines were read in Pass 1, producing 13 object records in Pass 2.
11 pages of virtual memory were used to define 10 macros.

+-----------------------------+
! Macro library statistics !
+-----------------------------+

Macro library name	Macros defined
_255DUA28:[SYSLIB]STARLET.MLB;2	7

200 GETS were required to define 7 macros.

There were no errors, warnings or information messages.

MACRO/ENABLE=SUPPRESSION/DISABLE=(GLOBAL,TRACEBACK)/LIS=LIS$:LIBASTINP/OBJ=OBJ$:LIBASTINP MSRC$:LIBASTINP/UPDATE=(ENH$:LIBASTINP)

RTLMACMAR
MAR

LIBASCEBC
LIS

LIBASTINP
LIS

RTLMACB32
REQ

STRMACROS
REQ

LIBBINTRE
LIS

RTLODBG
REQ

LIBDEF
FOR

LIBANASTR
LIS

LIBASNMBX
LIS

LIBBBCCI
LIS

LIBCHAR
LIS

LIBABUPCA
LIS

LIBADDX
LIS

RTLPSECT
REQ

SIGDEF
FOR

LIBASCTIM
LIS

LIBTABMAC
MAR

LIBATTACH
LIS

STRLNK
REQ

LIBA2EREV
LIS

LIBBBSSI
LIS

LIBCALLG
LIS

LIBCLICAL
LIS

LIBADDP
LIS